

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ
Декаан ФІКІС

Савенко О.С.
2021 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
Реінжиніринг та оптимізація програмних систем

Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Рівень вищої освіти – другий (магістерський)
Освітньо-професійна програма – інженерія програмного забезпечення
Шифр дисципліни – ОПП05
Статус дисципліни – обов'язкова
Факультет – Програмування та комп'ютерних і телекомунікаційних систем
Кафедра – інженерії програмного забезпечення

Форма навчання	Курс	Семестр	Обсяг дисципліни	Кількість годин						Курсовий проект	Курсова робота	Форма семестрового контролю	
				Кредити ЄКТС	Аудиторні заняття				Індивідуальна робота студента			Самостійна робота, в т.ч. ПРС	Залік
			Разом		Лекції	Лабораторні роботи	Практичні заняття						
Д	1	2	5	54	18	18	18		96			+	
Разом ДФН			5	54	18	18	18		96			+	

Робоча програма складена на основі освітньо-професійної програми.

Програма складена  к.т.н., доцент Форкун Ю.В.

 д.т.н., проф. Бармак О.В.

Схвалена на засіданні кафедри ПІЗ

Протокол № 1 від 31 08 2021 р.

Зав. кафедри ПІЗ  Бедратюк Л.П.
Підпис Іншталі, прізвище

Робоча програма розглянута та схвалена Вченою радою факультету програмування та комп'ютерних і телекомунікаційних систем

Голова Вченої ради


Підпис

Савенко О.С.
Ініціали, прізвище

Хмельницький 2021

РЕІНЖИНІРИНГ ТА ОПТИМІЗАЦІЯ ПРОГРАМНИХ СИСТЕМ

Тип дисципліни	Обов'язкова
Рівень вищої освіти	Другий (магістерський)
Мова навчання	Українська
Семестр	Перший
Кількість встановлених кредитів ЄКТС	5,0
Форми навчання, для яких викладається дисципліна	Денна

Результати навчання. Відповідно до Стандарту вищої освіти із зазначеної спеціальності та освітньої програми дисципліна має забезпечити: **компетентності**: Здатність генерувати нові ідеї (креативність). Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення. Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення. Здатність розробляти і реалізовувати наукові та/або прикладні проекти у сфері інженерії програмного забезпечення.. Здатність проектувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів. Здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення. Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення. Здатність критично осмислювати проблеми у галузі інформаційних технологій та на межі галузей знань, інтегрувати відповідні знання та розв'язувати складні задачі у широких або мультидисциплінарних контекстах. Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення. Здатність виконувати проектування програмних систем та їх складових компонентів із використанням поглиблених знань з оптимізації, реінжинірингу програмних систем, управління проектами та якістю з вибором парадигм програмування та застосування машинного навчання. **програмні результати навчання:** Розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника. Модифікувати існуючі та розробляти нові алгоритмічні рішення детального проектування програмного забезпечення. Приймати ефективні організаційно-управлінські рішення в умовах невизначеності та зміни вимог, порівнювати альтернативи, оцінювати ризики. Конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу. Прогнозувати розвиток програмних систем та інформаційних технологій. Здійснювати реінжиніринг програмного забезпечення відповідно до вимог замовника. Планувати, організувати та здійснювати тестування, верифікацію та валідацію програмного забезпечення. Вміти використовувати методи фундаментальних і прикладних дисциплін інженерії програмного забезпечення при проектуванні архітектури та розробці програмних систем із використанням поглиблених знань з оптимізації, реінжинірингу програмних систем, управління проектами та якістю з вибором парадигм програмування та застосуванням машинного навчання.

Зміст навчальної дисципліни. Продуктивність та якість програмного забезпечення. Реінженерія програмного забезпечення. Рефакторинг та оптимізаційні перетворення програмного забезпечення. Реінженерія та рефакторинг баз даних. Методи оптимізації процесу реінженерії програмних продуктів.

Запланована навчальна діяльність: лекції – 18 год., лабораторні заняття – 18 год., практичні заняття – 18; самостійна робота – 96 год., разом – 150 год.

Методи навчання: інтерактивні, пояснювально-ілюстративні, проблемного навчання і візуалізації, лабораторна робота, практична робота, індивідуальні завдання.

Форми і методи оцінювання результатів навчання: усне опитування, захист лабораторних робіт, тестовий контроль, теоретичне опитування

Вид семестрового контролю: іспит

Навчальні ресурси:

1. Martin Fowler. Refactoring: Improving the Design of Existing Code (2nd Edition). Addison-Wesley Professional, 2019. – 424p. ISBN 978-0134757599
2. Scott Ambler. Refactoring Databases: Evolutionary Database Design. Addison-Wesley Signature Series, 2017. – 384 p. ISBN 978-0-321-77451-4
3. Лаврищева Е.М. Программная инженерия и технологии программирования сложных систем. Учебник для вузов. – М. 2018. – 432 с.
4. Con Mike User Stories Applied for Agile Software Development. Addison-Wesley Professional, 2011. – 268 p. ISBN 0-321-20568-5
5. Модульне середовище для навчання MOODLE. Доступ до ресурсу: <https://msn.khnu.km.ua>.
6. Електронна бібліотека університету. Доступ до ресурсу: http://lib.khnu.km.ua/asp/php_f/page_lib.php.

Викладачі: доктор техн. наук, професор Бармак О.В., канд. техн. наук, доцент кафедри ІПЗ Форкун Ю.В.

3. ПОЯСНЮВАЛЬНА ЗАПИСКА

Мета викладання дисципліни: дисципліна «Реінжиніринг та оптимізація програмних систем» є однією із фахових дисциплін і займає провідне місце у підготовці фахівців освітнього рівня «магістр» за спеціальністю 121 «Інженерія програмного забезпечення» за освітньо-професійною програмою «Інженерія програмного забезпечення».

Мета дисципліни: формування компетентностей, необхідних для абстрактного мислення, аналізу та синтезу на відповідних рівнях оптимізації проектування та розробки програмних систем та їх компонентів; опанування студентами компонентних технологій та інтегрованих середовищ розробки для ефективної оптимізації програмного коду та програмних систем, конкретних виробничих задач інженерії програмного забезпечення.

Предмет дисципліни: Предметом вивчення дисципліни є реінжиніринг та оптимізація програмних систем.

Завдання дисципліни: Надати студентам знання і практичні навички практичного вирішення прикладних задач з обґрунтуванням реінжинірингу та оптимізації програмних систем

Відповідно до Стандарту вищої освіти із зазначеної спеціальності та освітньої програми дисципліна має забезпечити:

Компетентності: ЗК-1. Здатність до абстрактного мислення, аналізу та синтезу. ЗК-3. Здатність проводити дослідження на відповідному рівні. ФК-1. Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення. ФК-2. Здатність розробляти і реалізовувати наукові та/або прикладні проекти у сфері інженерії програмного забезпечення. ФК-3. Здатність проектувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів. ФК-5. Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення. ФК-6. Здатність ефективно керувати фінансовими, людськими, технічними та іншими проектними ресурсами у сфері інженерії програмного забезпечення. ФК-8. Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення. ФК-10 Здатність виконувати проектування програмних систем та їх складових компонентів із використанням поглиблених знань з оптимізації, реінжинірингу програмних систем, управління проектами та якістю з вибором парадигм програмування та застосування машинного навчання.

програмні результати навчання: ПРН-8 Розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника. ПРН-10 Модифікувати існуючі та розробляти нові алгоритмічні рішення детального проектування програмного забезпечення. ПРН12 Приймати ефективні організаційно-управлінські рішення в умовах невизначеності та зміни вимог, порівнювати альтернативи, оцінювати ризики. ПРН-13 Конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу. ПРН-14 Прогнозувати розвиток програмних систем та інформаційних технологій. ПРН-15 Здійснювати реінжиніринг програмного забезпечення відповідно до вимог замовника. ПРН-16 Планувати, організовувати та здійснювати тестування, верифікацію та валідацію програмного забезпечення. ПРН-18 Вміти використовувати методи фундаментальних і прикладних дисциплін інженерії програмного забезпечення при проектуванні архітектури та розробці програмних систем

із використанням поглиблених знань з оптимізації, реінжинірингу програмних систем, управління проектами та якістю з вибором парадигм програмування та застосуванням машинного навчання.

4. СТРУКТУРА ЗАЛКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

Назва теми	Кількість годин, відведених на:			
	Лекції	Лаб. роботи	Практичні роботи	СРС
Тема 1. Продуктивність, якість та оптимізація програмних систем.	2	2	2	12
Тема 2. Реінжиніринг програмного забезпечення.	4	4	4	12
Тема 3 Рефакторинг та оптимізаційні перетворення програмного забезпечення.	4	4	4	14
Тема 4. Реінжиніринг та рефакторинг баз даних.	4	4	4	14
Тема 5. Методи оптимізації процесу реінжинірингу програмних продуктів.	4	4	4	14
Разом	17*	17*	17*	96

Примітка.

* по чисельнику – 18 годин, по знаменнику – 16 годин (розрахунок здійснюється відповідно до розкладу занять)

5.ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

5.1. Зміст лекційного курсу

№ лекції	Перелік тем лекцій, їх анотації	К-сть годин
1	2	3
1	Продуктивність, якість та оптимізація програмних систем. Програмні та організаційні заходи визначення продуктивності та оптимізації ПЗ. Оптимізація етапів процесу розробки, їх закони, правила та умови проектування. Літ.: [1-6]	2
2	Основи реінжинірингу програмного забезпечення. Етапи та процеси реінжинірингу програмного забезпечення. Основні проблеми при плануванні задач реінжинірингу. Оптимізація робіт на етапі проектування програмної системи. Оцінка об'єму робіт на етапі проектування. Зворотня розробка програмного забезпечення. Основні задачі та етапи зворотної розробки. Ризики повторного використання коду програмного забезпечення. Літ.: [1-6]	2
3	Аналіз архітектури програмного забезпечення. Вимоги до проектів. Специфікації завдань Аналіз знань та алгоритмів. Специфікації завдань. Типи архітектур та їх моделі. Архітектури, засновані на потоках даних. Незалежні компоненти. Віртуальні машини. Репозиторні архітектури. Рівневі архітектури. Додатки зі змішаною архітектурою. Процедура вибору архітектури. Типи кодування. Літ.: [4,15]	2
4	Рефакторинг та оптимізаційні перетворення програмного забезпечення. Принципи рефакторингу. Формат методів рефакторингу. Організація та оптимізація програмного коду. Організація та оптимізація даних. [1,6]	
5.	Організація та оптимізація алгоритмів. Інструментальні засоби проведення рефакторингу. Літ.: [1,6]	2
6	Реінжиніринг та рефакторинг баз даних. Категорії операцій реінжинірингу та рефакторингу бази даних. Процеси реінжинірингу та рефакторингу бази даних Літ.: [2,5,14]	2
7	Операції реінжинірингу та рефакторингу бази даних. Операції реінжинірингу та рефакторингу структури бази даних [2,5,14]	
8.	Аналіз критеріїв ефективності програмних систем . Методи і моделі визначення ефективності використання програмних систем. Нечіткомножинні підходи й критерії оцінки програмних систем.[1-6,9-12] с. 22-32	2
9	Оптимізаційні моделі розробки програмного забезпечення з показниками величини проекту. Оптимізація на основі моделі заснованої на кількості помилок . Функція інтенсивності виявлення несправностей . Модель з індексом величини проекту. Процедури стохастичної оптимізації .Літ.: [1-6, 9-12,16] с. 363-395	2

	Разом за семестр	17
--	-------------------------	-----------

* по чисельнику – 18 годин, по знаменнику – 16 годин (розрахунок здійснюється відповідно до розкладу занять)

5.2. ЗМІСТ ЛАБОРАТОРНИХ ЗАНЯТЬ

№ з/п	Тема лабораторного заняття	Кількість годин
1	Лабораторна робота №1. Частина 1. Вимоги до проектів. Специфікації завдань. Умови оптимізації. Літ.: [1]	2
2	Лабораторна робота №1. Частина 2. Вимоги до проектів. Специфікації завдань. Умови оптимізації. Літ.: [1]	2
3	Лабораторна робота №2. Частина 1 Реінженерія архітектури програмного забезпечення. Літ.: [4,15]	2
4	Лабораторна робота №2. Частина 2 Реінженерія архітектури програмного забезпечення. Літ.: [4,15]	2
5	Лабораторна робота № 3. Частина 1. Рефакторинг та оптимізація програмного коду. Літ.: [1,6]	2
6	Лабораторна робота № 3. Частина 2. Рефакторинг та оптимізація програмного коду. Літ.: [1,6]	2
7	Лабораторна робота №4. Частина 1. Реінженерія та оптимізація БД. Оптимізація взаємодії програмної системи і СКБД. Літ.: [2,5]	2
8	Лабораторна робота №4. Частина 2. Реінженерія та оптимізація БД. Оптимізація взаємодії програмної системи і СКБД. Літ.: [2,5]	2
9	Підсумкове заняття	2
	Разом за семестр	17

* по чисельнику – 18 годин, по знаменнику – 16 годин (розрахунок здійснюється відповідно до розкладу занять)

5.3. ЗМІСТ ПРАКТИЧНИХ ЗАНЯТЬ

№ з/п	Тема лабораторного заняття	Кількість годин
-------	----------------------------	-----------------

1	Практична робота №1. Процеси розробки, їх закони, правила та умови проектування. Літ.: [1-4]	2
2	Практична робота №1. Процеси розробки, їх закони, правила та умови проектування. Літ.: [1-4]	2
3	Практична робота №2. Типи архітектур. Типи кодування. [4,15]	2
4	Практична робота №2. Типи архітектур. Типи кодування. [4,15]	2
5	Практична робота №3. Операції реінжинірингу та рефакторингу архітектури. Літ.: [1,4,15]	2
6	Практична робота №4. Методи та методики оптимізації коду.. Літ.: [1-4]	2
7	Практична робота №4. Операції рефакторингу якості даних.Літ.: [1,2,5]	2
8	Практична робота №4. Операції рефакторінгу бази даних Літ.: [2,5]	2
9	Підсумкове заняття	2
	Разом за семестр	17

* по чисельнику – 18 годин, по знаменнику – 16 годин (розрахунок здійснюється відповідно до розкладу занять)

5.3 ЗМІСТ САМОСТІЙНОЇ РОБОТИ

Об'єм самостійної роботи з дисципліни в 1 семестрі становить 96 годин. Самостійна робота студентів денної форми навчання полягає у систематичному опрацюванні програмного матеріалу, підготовці до виконання і захисту лабораторних робіт, тестування з теоретичного матеріалу, виконанні практичних завдань тощо.

Керівництво самостійною роботою здійснює викладач згідно з розкладом консультацій в позаурочний час.

Номер тижня	Зміст самостійної роботи	Кількість годин
1	2	3
1	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи № 1. Підготовка до виконання практичної роботи № 1.	2

17	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №4. Підготовка до виконання практичної роботи № 4.	2
18	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №4. Підготовка до виконання практичної роботи № 4.	
Разом		96

6. МЕТОДИ НАВЧАННЯ

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних методів: : інтерактивні, пояснювально-ілюстративні, проблемного навчання і візуалізації, лабораторна робота, практична робота, індивідуальні завдання і мають за мету – набуття студентами практичних навичок з реінжинірингу та оптимізації програмних систем за сучасними методиками.

7. ФОРМИ І МЕТОДИ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Поточний контроль здійснюється під час лекційних та лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочим планом дисципліни. Семестровий контроль проводиться у формі іспиту. При цьому при виведенні остаточної оцінки враховуються результати поточного контролю.

Використовуються наступні методи поточного контролю:

- усне опитування студентів перед допуском до лабораторного заняття;
- допуск до захисту лабораторних робіт;
- захист лабораторних робіт;
- тестування та перевірка працездатності створеного у ході виконання лабораторної роботи програмного, технічного та іншого забезпечення;

Підсумкова оцінка виставляється з урахуванням коефіцієнтів вагомості.

Студент, який не допущений до захисту або не захистив лабораторні роботи, вважається невстигаючим.

Процес оцінювання підготовленості студента можна розділити на етапи:

Перший етап оцінювання направлений на визначення знань інформаційного мінімуму. Якщо студент твердо засвоїв визначену навчальним планом суму формальних знань, то це означає, що він вміє використати їх при вирішенні різних питань при проектуванні виробів із шкіри, вміє розширити їх.

Перед вивченням дисципліни, як правило, проводиться вхідний контроль знань з дисциплін, що їй передують і забезпечують. При цьому необхідно встановити рівні та критерії сформованості знань щодо змісту навчальних елементів. Такими рівнями є:

Ознайомчо-орієнтовний (ОО) – особа має орієнтовне уявлення щодо понять, які вивчаються, здатна: відтворювати формулювання визначень поняття оптимізації програмних систем; орієнтуватись в методиках розробки програмного продукту; знати теоретичні основи створення додатків.

Понятійно-аналітичний (ПА) – особа має чітке уявлення щодо навчального об'єкту, здатна здійснювати смислове виділення, пояснення вибору конструктивних рішень щодо створення програмного продукту, методик їх проектування, їх обґрунтування на основі аналізу існуючих. Може чітко визначити структуру програм при їх проектуванні в залежності від їх використання, тобто здатна перенести раніше засвоєні знання на типові ситуації.

Продуктивно-синтетичний (ПС) – особа має глибоке розуміння щодо навчального об'єкту, здатна здійснювати синтез, генерувати нові ідеї та уявлення, переносити раніше засвоєні знання на нетипові, нестандартні ситуації. Тобто на цьому рівні студент повинен на основі теоретичних знань вміти проектувати різні види програмного продукту за найбільш поширеними методиками, вносити свої пропозиції щодо удосконалення способів проектування та програмування додатків; створювати конкурентоспроможні програмні додатки на світовому ринку.

При оцінюванні знань студентів використовуються різні засоби контролю, зокрема:

- допуск до виконання лабораторної роботи здійснюється на її початку усним опитуванням студента;
- якість виконання лабораторної роботи оцінюється шляхом перевірки виконаної лабораторної роботи у робочому файлі та в робочому зошиті;
- набуття теоретичних знань і практичних навичок перевіряється шляхом захисту кожної лабораторної роботи;
- засвоєння теоретичного матеріалу перевіряється тестовим контролем.

Оцінка, яка виставляється за лабораторну роботу, складається з таких елементів: усне опитування студентів перед допуском до виконання лабораторної роботи; знання теоретичного матеріалу з теми; якість оформлення результатів лабораторної роботи у зошиті та в робочому файлі; вміння студента обґрунтовувати прийняті рішення та робити висновки; своєчасний захист лабораторної роботи.

Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи занятті. За несвоєчасний захист лабораторної роботи (без поважної причини) захист лабораторної роботи оцінюється у три бали.

Пропущене з поважної причини лабораторне заняття студент повинен відпрацювати в лабораторіях кафедри в установленій викладачем термін.

Кожний вид роботи оцінюється за чотирибальною шкалою. Підсумкова оцінка з дисципліни виставляється як середньозважена з усіх видів робіт. Вагові коефіцієнти (ВК) видів робіт дисципліни приведено в таблицях 1-2.

Таблиця 1 - Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів денної форми навчання у семестрі за ваговими коефіцієнтами

Аудиторна робота				Самостійна, індивідуальна робота	Семестровий контроль, іспит
<i>Перший семестр</i>					
Лабораторні роботи №:				Тестовий контроль	Контрольна робота
Практичні роботи №					
1	2	3	4		Іспит
ВК*: 0,4				0,2	0,4

Умовні позначення: ВК – ваговий коефіцієнт.

Оцінка роботи студента за традиційною системою

Оцінка "відмінно" виставляється студентові, який вільно володіє матеріалом, чітко, грамотно, логічно, повно і послідовно висвітлює питання, демонструє знання не лише лекційного матеріалу, але й результати самостійного опрацювання літератури, вміння висвітлювати власні міркування та робити висновки.

Найвища оцінка ставиться студенту, який показав всебічне та глибоке знання навчально-програмного матеріалу, розбирається в поняттях, принципах та методах аналізу вимог до ПЗ, а також: який засвоїв основну і знайомий з додатковою літературою, засвоїв основні поняття дисципліни; дав повні і точні відповіді на всі запитання з поясненнями і висновками; проявив творчі здібності в розумінні, викладенні і використанні навчального матеріалу.

Оцінка "*добре*" виставляється студентові, який добре володіє навчальним матеріалом, грамотно і логічно викладає його в усній чи письмовій формі, допускаючи при цьому не більше одного-двох незначних помилок, Обмежується викладом основних теоретичних положень, не володіє матеріалом додаткової літератури, хоча загалом має тверді знання.

"*Задовільно*" отримує студент, що показав знання основного матеріалу в обсязі, необхідному для подальшого навчання (основні поняття, принципи, закономірності), а також: знайомий з основною літературою; при відповідях на запитання відчуває себе невпевнено, допускає незначні помилки; при допусканні погрешностей у відповідях володіє знаннями для їх виправлення під керівництвом викладача.

Оцінка "*незадовільно*" виставляється студентові, який демонструє не зовсім повні знання курсу навчальної дисципліни чи її окремої теми, допускає при викладанні навчального матеріалу суттєві помилки, не орієнтується у лекційному матеріалі та матеріалі основної і додаткової літератури.

"*Незадовільно*" ставиться студентові, який має пробіли в знаннях основного матеріалу, а також: допустив принципові помилки, не розуміє основних концептуальних положень дисципліни; на запитання не дає задовільних відповідей, не справляється з рішенням задач; не може продовжувати навчання без додаткових занять з дисципліни.

Якщо студент отримав негативну оцінку, то він має перездати її в установленому порядку, але обов'язково до терміну наступного контролю. У випадку, коли студент не виконав індивідуальний план з дисципліни у заплановані терміни без поважних причин, то під час відпрацювання заборгованості при позитивній відповіді йому виставляється оцінка „задовільно”.

Підсумкова семестрова оцінка за національною шкалою і шкалою ЄКТС встановлюється в автоматизованому режимі після внесення усіх оцінок до електронного журналу. Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС наведені у табл. 2.

Залік виставляється, якщо середньозважений бал, який отримав студент з дисципліни, знаходиться в межах від 3,00 до 5,00 балів. При цьому за вітчизняною шкалою ставиться «зараховано», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом кількості балів. Перехід від вітчизняної шкали оцінювання до європейської (ECTS) представлено у таблиці 2:

Таблиця 2 - Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ECTS	Бали	Вітчизняна система	
A	4,75-5,00	5	ВІДМІННО - глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навичок
B	4,25-4,74	4	ДОБРЕ - повне знання навчального матеріалу з кількома незначними помилками
C	3,75-4,24	4	ДОБРЕ - в загальному правильна відповідь з двоматрьома суттєвими помилками

D	3,25-3,74	3	ЗАДОВІЛЬНО - неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією
E	3,00-3,24	3	ЗАДОВІЛЬНО - неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00-2,99	2	НЕЗАДОВІЛЬНО - безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00-1,99	2	НЕЗАДОВІЛЬНО - необхідна серйозна подальша робота і повторне вивчення дисципліни

8. ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ СТУДЕНТІВ

1. Ознаки продуктивності програмного забезпечення.
2. Надійність програмних систем.
3. Програмні та організаційні заходи для визначення продуктивності ПЗ.
4. Процеси розробки, їх закони, правила та умови проектування.
5. Реінжиніринг програмних систем
6. Вимоги до проектів.
7. Специфікації завдань.
8. Типи архітектур.
9. Типи кодування.
10. Мови, типи платформ, кросплатформеність.
11. Оточення середовище програмування і проектування.
12. Фактори процесу проектування програмного продукту.
13. Оптимальне конструювання модулів програмних систем
14. Продуктивність у суміжних областях ІТ.
15. Рефакторинг програмних систем
16. Оптимізація постійних виразів.
17. Організація та оптимізація даних.
18. Організація та оптимізація програмного коду.
19. Організація та оптимізація алгоритмів.
20. Частота оновлення.
21. Частота фіксації помилок.
22. Заміна змінних константними значень ("розмноження" констант).
23. Вирахування значень змінних на стадії компіляції (константа "згортання").
24. Вирахування значень функцій на стадіях компіляції ("згортка" функцій).
25. Оптимізація алгебраїчних виразів.
26. Видалення невикористовуваних змінних.
27. Видалення копій змінних.
28. Видалення невикористовуваних присвоєвань.
29. Видалення зайвих виразів.
30. Видалення зайвих викликів функцій.
31. Виконання алгебраїчних спрощень.
32. Основні методи оптимізації.

33. Оптимізація підвиразів.
34. Оптимізація арифметичних операцій.
35. Оптимізація розгалужень.
36. Заміна переходів арифметичних операцій.
37. Видалення зайвих умов.
38. Видалення завідомо хибних умов.
39. Оптимізація switch.
40. Балансування логічного дерева.
41. Створення таблиць переходів.
42. Оптимізація циклів.
43. Оптимізація викликів функцій.
44. Оптимізація передачі аргументів.
45. Оптимізація функцій. Оптимізація розподілу змінних.
46. Оптимізація ініціалізації строк.
47. Оптимізація "мертвого" коду. Оптимізація константних умов.
48. Реінжиніринг та оптимізація даних у СКБД.
49. Старіння процедурного кешу.
50. Неоптимальні індекси.
51. Неоптимальні статистики.
52. Оптиміальне конструювання модулів програмних систем.
53. Критерії оцінки якості машиної оптимізації.
54. Методики оцінки якості машиної оптимізації.

9. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Навчальний процес з дисципліни «Реінжиніринг та оптимізація програмних систем» забезпечений необхідною навчально-методичною літературою. Підготовлено курс у модульному середовищі ХНУ.

10. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна література:

1. Martin Fowler. Refactoring: Improving the Design of Existing Code (2nd Edition). Addison-Wesley Professional, 2019. – 424p. ISBN 978-0134757599
2. Scott Ambler. Refactoring Databases: Evolutionary Database Design. Addison-Wesley Signature Series, 2017. – 384 p. ISBN 978-0-321-77451-4
3. Con Mike User Stories Applied for Agile Software Development. Addison-Wesley Professional, 2011. – 268 p. ISBN 0-321-20568-5
4. Len Bass, Paul Clements, Rick Kazman. Architektura oprogramowania w praktyce. Wydanie II. Helion. 2011, - 464 s. ISBN 978-83-246-3302-9
5. Scott W. Ambler and Pramod J. Sadalage. Refactoring Databases. Addison-Wesley, 2006. – 388p. ISBN: 0321293533
6. Зайцев. Є. О. Основи програмної інженерії: навчальний посібник. К.: КНТЕУ, 2017. – с.

7. Управління проектами: навчальний посібник до вивчення дисципліни для магістрів галузі знань 07 Управління та адміністрування / Уклад.: Л.Є. Довгань, Г.А.Мохонько, І.П.Малик. – К.: КПІ ім. Ігоря Сікорського, 2017. – 420 с. [Електронний ресурс] – Режим доступу: <http://ela.kpi.ua/handle/123456789/19481>.

Додаткова література:

1. Sridhar B. Model Driven Software Engineering in the Mobile Era with an Emphasis on Security / International Journal of Electronics Communication and Computer Engineering. Vol. 6, Iss. 6. – 2015. – P. 619-623.
2. Сазерленд Д. Scrum. Навчись робити вдвічі більше за менший час / Д. Сазерленд; пер. з англ. Я. Лебеденко. – Х.: Книжковий Клуб «Клуб Сімейного Дозвілля», 2016.– 281с.
3. Грабовецький Б. Є. Методи експертних оцінок: теорія, методологія, напрямки використання: монографія //Вінниця: ВНТУ. – 2010. – Т. 171.
4. Рач В.А. Кількісні методи ресурсно-часового оцінювання в гнучкій методології розробки програмного забезпечення / В. А. Рач, А. І. Пилипенко // Управління проектами та розвиток виробництва. – 2017. – № 1. – С. 62-71. – Режим доступу: http://nbuv.gov.ua/UJRN/Uprv_2017_1_7

11. ІНФОРМАЦІЙНІ РЕСУРСИ

1. Модульне середовище для навчання. Доступ до ресурсу: <https://msn.khnu.km.ua/mod/quiz/view.php?id=232713>.
2. Електронна бібліотека ХНУ. Доступ до ресурсу: http://lib.khnu.km.ua/asp/php_f/p1age_lib.php.
3. Репозитарій ХНУ. Доступ до ресурсу: <http://elar.khnu.km.ua/jspui/?locale=uk>.